Paradigma Lógico

Aula 08

Profª. M. Sc. Luciana De Nardin

Listas em Prolog

- São estruturas simples de dados, largamente empregadas em computação não-numérica
- É uma sequência ordenada e pode ter qualquer comprimento
- Todos os objetos em Prolog são árvores → listas também são!
- Representação:

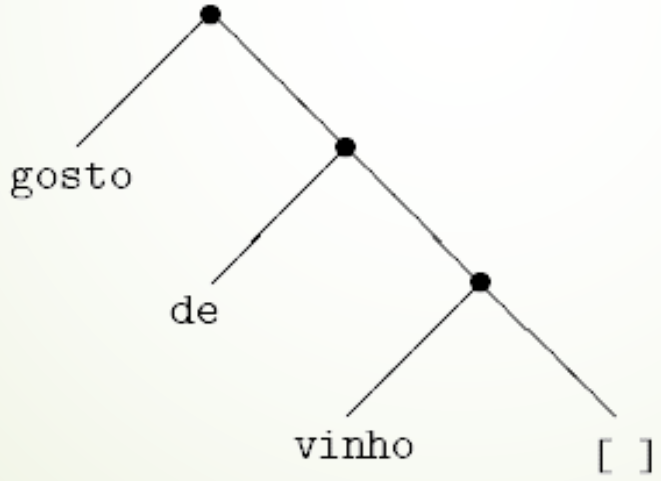
```
[riodejaneiro, saopaulo, pocosdecaldas]
```

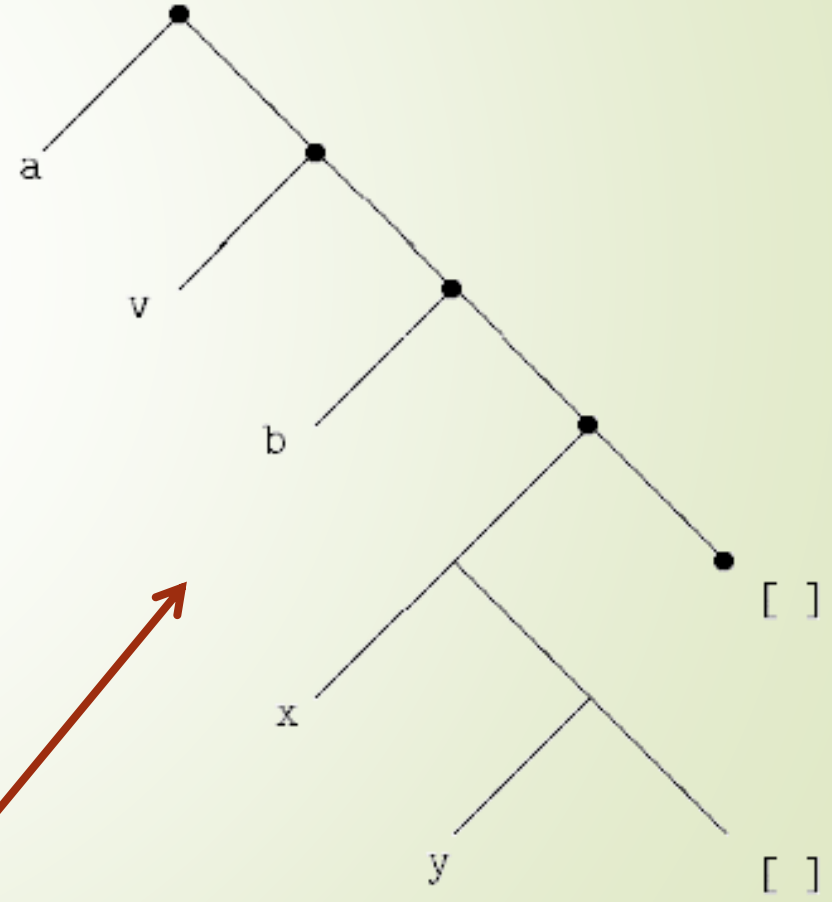
Sintaxe de Listas

- Lista vazia: representada pelo átomo []
- Lista não vazia: deve ser pensada com dois componentes → cabeça e cauda
- A lista que tem cabeça e cauda é denotada por [X | Y]
- Dentro de uma lista os elementos são separados por vírgula (,)

Sintaxe de Listas

Lista	Cabeça	Cauda
[gosto,de,vinho]	gosto	[de,vinho]
[[3],5,[2,7]]	[3]	[5,[2,7]]
[X,Y|Z]	X	[Y|Z]
[[o,gato]]	[o,gato]	[]





[a,v,b,[x,y]]

Unificação em Listas

- Regras para unificação:
  - Se S e T forem constantes, então, S e T unificam, se e somente se, forem o mesmo objeto
  - Se S for uma variável e T qualquer termo, então unificam e S é instanciado com T; vice-versa, a variável T é instanciada com S

Unificação em Listas

- `? - data (D, maio, 2002) = data(25,maio,1983).`

Não unifica: constantes numéricas diferentes

- `? - data (D, maio, 1983) = data(25,maio,Ano).`

```
D = 25
Ano = 1983;
no
```

Unificação em Listas

- Regras para unificação:
  - Se S e T são estruturas, elas unificam, se e somente se, todos os elementos correspondentes unificam

```
[X|Y] = [a,b,c,d]
```

```
X = a
Y = [b,c,d]
```

Unificação em Listas

- Regras para unificação:
  - Se os termos não se unificam diz-se que o processo de unificação falha
  - Se os termos se unificam, diz-se que o processo de unificação é bem sucedido e as variáveis em ambos os termos são instanciadas com valores que tornam os termos idênticos

Unificação em Listas

Lista 1	Lista 2	Unificação
[mesa]	[X|Y]	X/mesa Y/[ ]
[a,b,c,d]	[X,Y|Z]	X/a Y/b Z/[c,d]
[[ana,Y]|Z]	[[X,foi],[ao,cinema]]	X/ana Y/foi Z/[[ao,cinema]]
[ano,bissexto]	[X,Y|Z]	X/ano Y/bissexto Z/[]
[ano,bissexto]	[X,Y,Z]	não unifica
[data(7,Z,W),hoje]	[X|Y]	X/data(7,Z,W) Y/[hoje]
[data(7,W,1993),hoje]	[data(7,X,Y),Z]	X/W Y/1993 Z/hoje

Recursividade em Listas

- Considere a seguinte lista dos nomes dos filhos de Júpiter, rei do Olimpo e de sua esposa Juno:

```
[ilitia, marte, hebe, vulcano]
```

- Para verificar se um dado nome está na lista, deve-se verificar se ele é cabeça da lista ou se ele está na cauda da lista
- Se o final da lista for atingido, significa que a busca não foi bem sucedida

Recursividade em Listas

```
pertence(Elem,[Elem|_]).
pertence(Elem,[_|Cauda]):- pertence(Elem,Cauda).
```

Obs.:

- "_" é interessante para evitar que o Prolog realize tarefas desnecessárias.
- Vale também utilizá-lo quando não é necessário que o Prolog se lembre das instanciações feitas nesta unificação

Recursividade em Listas

- Testes:
  - `? - pertence(a,[l,a,b]).`
  - `? - pertence(a,[hoje,ontem]).`
  - `? - pertence(X,[a,b,c,d,e]).`

  - `? - pertence(a,X).`
  - `? - pertence(X,Y).`

Exercícios

- Defina predicados para:
  - Retornar o último elemento de uma lista
  - Verificar se dois elementos são consecutivos em uma lista
  - Somar todos os elementos numéricos de uma lista

Exercícios (resolução)

- Último elemento de uma lista
  - Se a lista tem apenas um elemento, este elemento é seu último elemento
  - O último elemento de uma lista com mais de um elemento é o último elemento da cauda da lista

Exercícios (resolução)

- Dois elementos consecutivos
  - Dois elementos E1 e E2 são consecutivos em uma lista se eles forem o primeiro e o segundo elementos da lista
  - Ou, se forem consecutivos na cauda da lista

Exercícios (resolução)

- Soma dos elementos de uma lista
  - Se a lista for vazia, a soma dos elementos é zero
  - Se a lista for [Elem|Cauda] a soma de seus elementos é a soma dos elementos da Cauda mais o primeiro elemento Elem